UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/672,368	09/28/2000	Francis X. McKeen	042390.P9575	7652

7590 07/06/2007

Blakely Sokoloff Taylor & Zafman LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025

EXAMINER	
HO, THOMAS M	

ART UNIT	PAPER NUMBER
2132	

MAIL DATE	DELIVERY MODE
07/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

MAILED

JUL 06 2007

Technology Center 2100

Application Number: 09/672,368
Filing Date: September 28, 2000
Appellant(s): MCKEEN ET AL.

For Appellant
Gregory Caldwell
Reg 39,926

EXAMINER'S ANSWER

This is in response to the appeal brief filed 1/10/2007 appealing from the Office action mailed 9/6/2006.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6098133	Summers et al.	8-2000
5615263	Takahashi	3-1997
5729760	Poisner	3-1998

- Coulouris et al. pages "Distributed Systems Concepts and Designs", 1994, pgs 165-194, 300-308
- Silberschatz et al. "Operating System Concepts", 5th edition, 1999, pgs 94, 264, 267, 270-272, 289, 293, 402-405, 444-445,

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Coulouris et al. and Silberschatz et al. and Summers et al., US patent 6098133.

In reference to claim 1:

(Coulouris et al. Section 6.3 Processes and Threads) discloses a method comprising:

- Identifying if an event is one of a class of events to be handled in the isolated execution mode, where the isolated execution mode is a processor running a secure process (Page 168), and the event is one of an event or events that might be handled by that process, where threads within a process have their own software interrupt handling mechanisms
- Handling the event using the first page table map if the event is identified as one of the class of events to be handled by the isolated execution mode, where the first page table map is the virtual memory map which maps the memory for the running processes(page

169, 190-192), and the event identified as one of the events to be handled by the isolated execution mode is an event that is to be handled by that process. (page 172)

Coulouris et al. does not explicitly disclose

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode.
- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode.

Silberschatz et al. (p 270-271) discloses

- Maintaining a first page table map for use in an isolated execution mode and a second page table map for use in a normal execution mode, where the first page table map is a standard process which executes its own code in an isolated manner, and the normal execution mode is the special case of shared pages between processes.
- Dynamically swapping between the first page table map and the second page table map responsive to a change in execution mode, where processes are isolated execution modes and changing from one execution mode to another would involve a context switch from one process that doesn't use shared pages to another that does. P. 92 (processes)

Silberschatz et al. (p 270-271) discloses that there is an advantage to sharing common code, particularly in the context of a time-sharing environment, and that reentrant shared code can result in a significant savings of total memory space. P. 271 (paragraph 2)

Neither Silberschatz et al. or Coulouris et al. explicitly recites the limitation

- Restricting access to an isolated area of memory to bus cycles performed in the isolated execution mode.

However Silberschatz et al. or Coulouris et al. do disclose restricting access to an isolated area of memory.

A bus, is merely the path that connects the various components of a computer to allow data to be transferred from one internal component to another. All Buses transfer data in cycles as a synchronous device.

Summers et al. discloses

- Restricting access to an isolated area of memory to bus cycles performed in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the shared code processes of Silberchatz et al. with the isolated execution processes of Coulouris et al. in order to allow for significant savings in memory while still retaining the logical boundaries of the process to allow for managed concurrent execution and to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 2:

(Coulouris et al. Section 6.3 Processes and Threads) discloses the method of claim 1 further comprising:

- Identifying if the event is one of a class of events to be handled in the isolated execution mode, where the isolated execution mode is a processor running a secure process (Page 168), and the event is one of an event or events that might be handled by that process, where threads within a process have their own software interrupt handling mechanisms
- Handling the event using the first page table map if the event is identified as one of the class of events to be handled in the isolated execution mode, where the first page table map is the virtual memory map which maps the memory for the running processes(page 169, 190-192), and the event identified as one of the events to be handled by the isolated execution mode is an event that is to be handled by that process. (page 172)
- Wherein identifying comprises indexing into a lookup table with a exception vector of the event, where the identifying of the interrupt comprises indexing the disclosed lookup

table Silberschatz et al. page (404) with the interrupt or "exception" vector page (403) & Silberschatz et al. page (402-404)

In reference to claim 3:

Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein dynamically swapping comprises:

- Loading a set of control registers selected based on an exception vector of the event, where a set control registers may be found with the data loaded from the interrupt descriptor table registers in the case of an event, where the control registers are the memory addresses of specialized interrupt handlers which are controlled by the event (exception) table. Silberschatz et al. page (402-404)

In reference to claim 4:

Coulouris et al. and Silberschatz et al. fail to explicitly disclose the method of claim 3 wherein the set of control registers comprises:

- A global descriptor table register
- An interrupt descriptor table register
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention. In particular a

GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.
Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

In reference to claim 5:

Coulouris et al. and Silberschatz et al. discloses the method of claim 1 wherein maintaining comprises:

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

Silberschatz et al.(page 445) discloses a RAID organization called mirroring in which the whole disk is duplicated. While costly, the advantages of this allow reading that is twice as fast. Silberschatz et al(p. 289) also discloses that memory maps, page tables, and processes may be placed on the actual hard disk itself in virtual memory. Silberschatz et al. discloses on p. 293, Figure 9.3 that page tables and memory maps for the memory may be stored in the actual hard disk.

The mirroring a hard disk containing virtual memory on it as disclosed by Silberschatz et al. inherently discloses

- Mirroring a page table base address register.
- Mirroring a memory map is not explicitly disclosed however,

In reference to claim 6:

(Coulouris et al. Section 6.4 Naming and Protection) discloses the method of claim 1 further comprising:

- Defining a set of events that should be handled in isolated execution mode, where the set of events that should be handled by the isolated execution mode are the set of events that should be handled by a particular running process, selected by the server.

In reference to claim 7:

(Coulouris et al. Section 10.4 Distributed Coordinarion) discloses the method of claim 6 wherein the set of events to be handled in the isolated execution mode comprises:

machine check events and clock events, where the machine and clock events involve the synchronization of system clocks in a distributed system.

In reference to claim 8:

Coulouris et al. discloses the method of claim 2 wherein handling comprises:

- Determining if a current mode is the isolated execution mode, where the current mode is determined if it is in isolated execution mode, if it is determined that an isolated process is currently running. (Section 6.4 Naming and Protection)
- Loading a set of control registers with values corresponding to the first page table map if the current mode is not the isolated execution mode and the event is one of the class, where the set of control registers are loaded which contain the descriptor for the set of instructions needed to handle the current event, if it is found that the event is not to be handled by the current running process, but by another process. (Section 6.4 Naming and Protection)
- Dispatching an exception vector after the loading is complete, where the exception vector for the event is be dispatched once the new process capable of handling the event is loaded or switched to. (Section 6.4 Naming and Protection) & Figure 6.12

Claims 9-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Takahashi, US Patent 5615263 in view of Summers et al. , US patent 6098133.

In reference to claim 9:

Takahashi discloses an apparatus comprising:

- A first storage location storing control data for a first page table map, where the first page table map is the map that designates the memory. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)
- A second storage location storing control data for a second page table map, where the second storage location is the ROM. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)
- A selection unit to select which page table map is applied responsive to receipt of an event, where the selection unit chooses to select between the ROM and the memory based on the execution mode of the processor. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40) & (Column 2, lines 45 – 61)

Takahashi fails to explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode.

Summers et. al. discloses

- An isolated execution circuit to generate isolated access bus cycles, (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54) &

(Column 2, line 60 – Column 3, line 15), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 10:

Takahashi and Summers et al. discloses the apparatus of claim 9 wherein the selection unit comprises:

- A multiplexer that selects between the first and second storage locations based on an exception vector of the event. (Figure 1, Item 13) & (Column 2, line 62 – Column 3, line 15)

In reference to claim 11:

Takahashi and Summers et al. (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40) & (Column 2, lines 45 – 61) discloses the apparatus of claim 9 wherein the first storage location contains a base address for the first page table map and the second storage location contains a base address for the second page table map.

In reference to claim 12:

Takahashi discloses a platform comprising:

- A processor executing in one of a normal execution mode and isolated execution mode; (Column 2, lines 45 – 61)
- A first set of control registers to define a current memory map of the platform; (Column 3, lines 45-60)
- A mapping unit to dynamically load the first set of control registers responsive to an event if the event should be handled using an alternative memory map; (Figure 5) & (Column 3, lines 45-60) & (Column 4, lines 23-60) & (Column 3,lines 25-40)

Takahashi fails to explicitly disclose

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode.

Summers et al. discloses

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54),

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 13:

Takahashi and Summers et al. discloses the platform of claim 12 wherein the mapping unit comprises:

- A second set of registers having a first subset corresponding to control register values for a normal execution mode memory map and a second subset corresponding to control register values for an isolated execution mode memory map, where the isolated memory

map is the ROM, read only memory containing the secure functions. (Column 3, lines 60 Column 4, lines 60)

- A selection unit to select between the first subset and the second subset. (Column 3, lines 25-47)

In reference to claim 14:

Takahashi and Summers et al. discloses the platform of claim 13 wherein the selection unit comprises:

- A multiplexer having selection driven by an exception vector of an incoming event. (Figure 1, Item 13) & (Column 2, line 62 – Column 3, line 15)

However the use of multiple multiplexers is not explicitly disclosed.

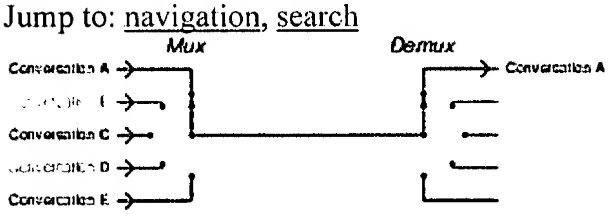
The Examiner takes official notice that using a plurality of multiplexers as opposed to a single multiplexer was well known in the art at the time of invention.

Multiplexer

From Wikipedia, the free encyclopedia

(Redirected from Multiplexor)

Jump to: navigation, search



The basic function of a multiplexer: combining multiple inputs into a single data stream. On the receiving side, a demultiplexer splits the single data stream into the original multiple signals.
A **multiplexer** (or **mux** or, more rarely, **muldex**) is an encoder that combines two or more inputs into a single output. In electronics, the multiplexer combines several electrical signals into a single signal. There are different types of multiplexers for analog and digital circuits.
In digital signal processing, the multiplexer takes several separate digital data streams and combines them together into one data stream of a higher data rate. This allows multiple data streams to be carried from one place to another over one physical link, which saves cost.

In fact, multiple multiplexers may be used without any change to the input and output of a digital system as opposed to a single multiplexer if arranged to be logically equivalent. It would have been obvious to one of ordinary skill in the art at the time of invention to use multiple multiplexers to combine different size data streams into a single larger data stream.

In reference to claim 15:

Takahashi and Summers et al. fails to explicitly disclose the platform of claim 12 wherein the first set of control registers comprises:

- A global descriptor table register;
- An interrupt description table register;
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention as part of a processor. In particular a GDTR and an IDTR are registers that contain entries which associate

each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.

Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

Claims 9-15 are further rejected under 35 U.S.C. 103(a) as being unpatentable over Poisner, US Patent 5729760 in view of Summers et al. , US patent 6098133.

In reference to claim 9:

Poisner discloses an apparatus comprising:

- A first storage location storing control data for a first page table map, where the first table map is the unrestricted memory as indicated by the IO mapped register. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A second storage location storing control data for a second page table map, where the second table map is the restricted memory as indicated by the IO mapped register. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

- A selection unit to select which page table map is applied responsive to receipt of an event, where the selection unit makes the determination based on the mode of processor execution. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

Poisner fails to explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode.

Summers et. al. discloses

- An isolated execution circuit to generate isolated access bus cycles, (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54), where the access to the memory via the bus are also restricted with a secure bus mechanism.
- Wherein isolated access bus cycles are to be used if the apparatus operates in an isolated execution mode (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54) & (Column 2, line 60 – Column 3, line 15), where the access to the memory via the bus are also restricted with a secure bus mechanism.

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his

invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 10:

Poisner (Column 8, line 42 – Column 9, line 15) discloses the apparatus of claim 9 wherein the selection unit comprises:

- A multiplexer that selects between the first and second storage locations based on an exception vector of the event.

In reference to claim 11:

Poisner (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67) discloses the apparatus of claim 9 wherein the first storage location contains a base address for the first page table map and the second storage location contains a base address for the second page table map.

In reference to claim 12:

Poisner discloses a platform comprising:

- A processor executing in one of a normal execution mode and isolated execution mode, where the normal mode of execution is the mode of execution that uses the unrestricted IO map and the isolated mode of execution uses the restricted IO map. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A first set of control registers to define a current memory map of the platform, where the IO memory maps are defined in the control registers. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A mapping unit to dynamically load the first set of control registers responsive to an event if the event should be handled using an alternative memory map, where the memory map are the different memories accessed depending on the different modes of execution for the processor, and each memory map is dynamically loaded based on the mode of the processor. (Figure 8) & (Figure 9) & (Figure 10) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

Poisner does not explicitly disclose:

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode.

Summers et al. discloses

- An isolated execution circuit to generate isolated access bus cycles if the processor is executing in the isolated execution mode. (abstract) & (Column 2, lines 47-61) & (Column 3, lines 38-54),

Summers et al. discloses that providing an isolated path needs to be established for transmitting certain data to ensure that the data is received by authorized recipients, and that unauthorized elements have not been intercepted. (Column 1, lines 15-28) Summers et al. teaches that his invention provides an advantage over other secure bus lines by providing a secure bus arbiter module that is useable in any commercial off the shelf motherboard. (Column 1, lines 50-56)

It would have been obvious to one of ordinary skill in the art at the time of invention to use the secure bus arbiter of Summers et al. to ensure that data may be transferred securely from one module to another within the computer in a way that is compatible with off the shelf, common motherboards.

In reference to claim 13:

Poisner discloses the platform of claim 12 wherein the mapping unit comprises:

- A second set of registers having a first subset corresponding to control register values for a normal execution mode memory map and a second subset corresponding to control register values for an isolated execution mode memory map; (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)
- A selection unit to select between the first subset and the second subset. (Figure 8) & (Column 2, lines 55 – Column 3, lines 52) & (Column 4, lines 42-67)

In reference to claim 14:

Poisner (Column 8, line 42 – Column 9, line 15) discloses the platform of claim 13 wherein the selection unit comprises:

- A multiplexer having selection driven by an exception vector of an incoming event.

However the use of multiple multiplexers is not explicitly disclosed.

The Examiner takes official notice that using a plurality of multiplexers as opposed to a single multiplexer was well known in the art at the time of invention.

Multiple multiplexers may be used without any change to the input and output of a digital system as opposed to a single multiplexer if arranged to be logically equivalent. It would have been obvious to one of ordinary skill in the art at the time of invention to use multiple multiplexers to combine different size data streams into a single larger data stream.

In reference to claim 15:

Poisner fails to explicitly disclose the platform of claim 12 wherein the first set of control registers comprising:

- A global descriptor table register;
- An interrupt description table register;
- A page table map base address register.

The examiner takes as admitted prior art that a global descriptor table register and an interrupt descriptor table register were well known in the art at the time of the invention as part of a processor. In particular a GDTR and an IDTR are registers that contain entries which associate each interrupt or exception identifier with a descriptor for the set of instructions that are to service the event.

Both of these registers are disclosed in a number of processors and processor programming manuals include the well known 80386 Programmer Reference Manual.

- It would have been obvious to one of ordinary skill in the art at the time of invention to have a GDT register and an IDT register, so that processor knows which set of instructions to use to respond to a particular event.

(10) Response to Argument

There are several concepts of the art pertinent to the present grounds of rejection that will provide clarification in its understanding.

Process:

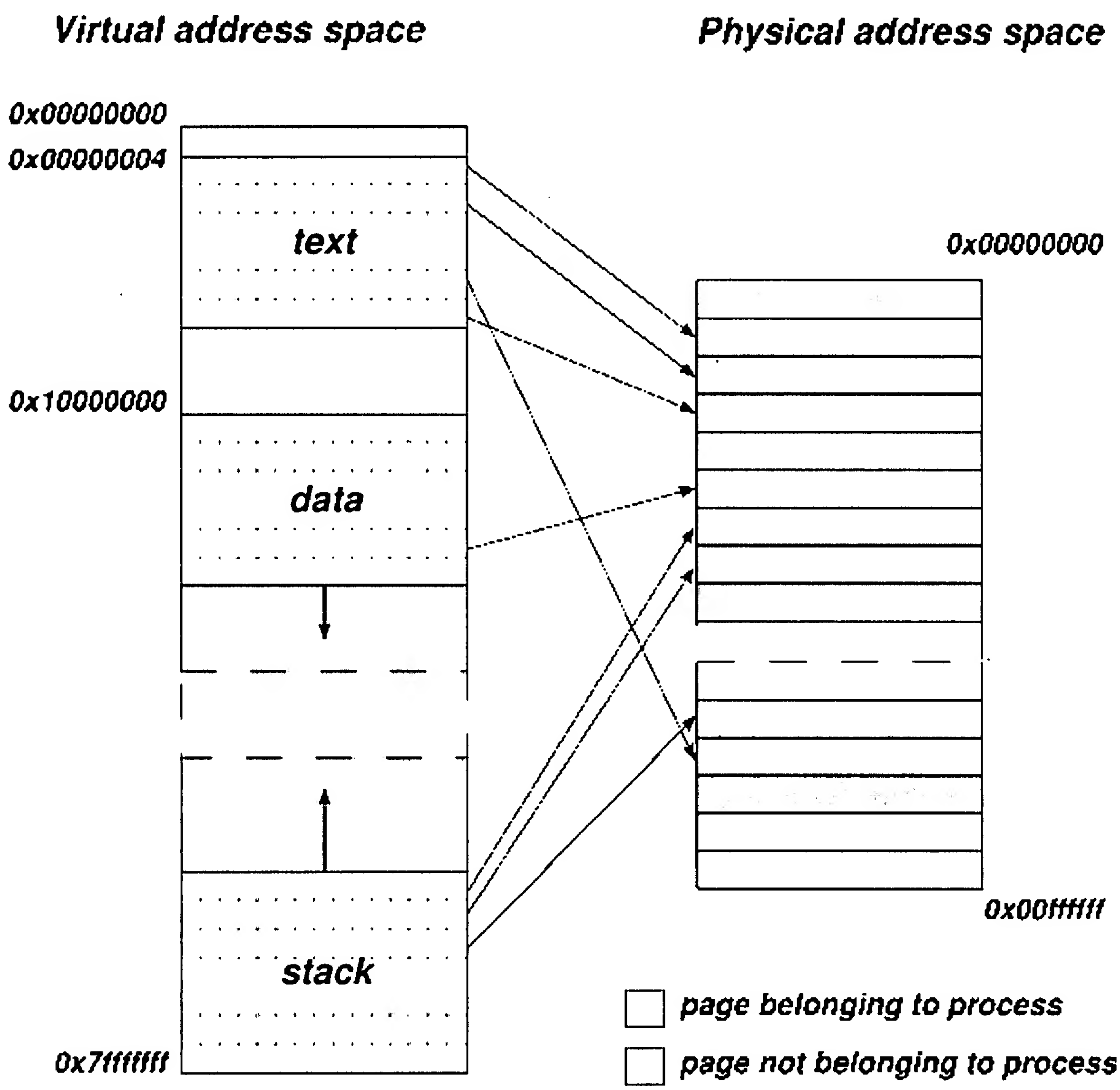
All major executing programs within the context of an operating system for use in a digital system are known as "processes." The process is a logical construct which provides a logical division between concurrent executing programs on a computer.

For example, if the Examiner were to open up a web browser and word editor program, these programs would execute as running processes. A process as known in the art contains its own specific allocated memory and resources to provide the context within which it is supposed to run. A running process cannot typically access the memory allocated to another process. In this respect processes are "isolated" from one another. Additionally, because each process maintains its own memory and resources, each process has inherent to it, a mapping regarding its memory structure and which addresses of the memory are allocated to the process.

Page table:

A page table as understood in the art is the data structure used to map virtual addresses to physical addresses. For example, if a process is allocated 1024K of memory, the process may use this memory as if it were a singular 1024 chunk. However, the actual physical addresses may not be consecutive, but may be scattered throughout the physical memory structure. (a byte here and a byte there)

A scattered, nonconsecutive set of addresses of memory is mapped to a logical construct in a mapping known as a page table mapping. For example:



isolated execution as is the execution of specific code or set of computer instructions in isolation. The isolation limits the execution to the extent that other programs, software, and/or hardware is somehow restricted in its access to the memory of set of instructions executing in isolation.

The fundamental basis behind methods of implementing "isolated execution" primarily lies in isolating areas of memory.

Response to argument:

In reference to claim 1:

No different bus cycles are used whether the page table entries are shared or not. Thus even if Coulouris and Silberschatz did teach or suggest the various claim limitations as the Examiner alleges, there is no "hook" (explicit or implicit) to tie in an analysis of the bus cycles. Appeal Brief (page 6, paragraph 3)

Summers describes an apparatus for isolating an untrusted peripheral from the bus of a computer when data that the peripheral should not see or affect is passing over the bus. Figure 7 shows clearly where the apparatus resides. : "the secure bus arbiter" 71 is placed between the non-secure card 72 and the system backplane 75.

...

Even assuming that these signals caused or were associated with bus cycles performed in a particular execution mode(they do not, and are not), the apparatus isolates an untrusted peripheral. To does not restrict access to an isolated area of memory. Appeal Brief (page 7 paragraph 2)

The Examiner has not identified any mechanism for restricting access to an isolated area of memory to bus cycles performed in the isolated execution mode. All of Summer's bus cycles seem to be the same; the difference is whether the untrusted peripheral can observe the bus during a bus cycle.

The Appellant appears to argue that the combination rendered for the rejection of claim 1 does not disclose the limitation of restricting access to an isolated area of memory to bus cycles.

The Examiner contends however all computer buses have bus cycles. A bus is the pathway through which information in passed around internally to a computer system. A bus cycle is

merely the unit of time in which the bus performs a single transaction. It is analogous to a CPU clock cycle.

Summers discloses a secure bus arbiter arrangement in which a secure bus is used to maintain isolation between different classes of data. (Column 2, lines 45-65)

In response to the Appellant's contention that Summer's bus cycles seem to be the same, the Examiner contends that (Column 2, lines 45-65) recites that the buses must incorporate logic that permits their association with certain classes of data and prohibits their association with other classes.

As recited in the rejection of claim 1, Summer's additionally provides support in (Column 3,lines 38-54) by stating that the secure bus arbiter is able to open and close various bus lines to establish a secure line when necessary.

Thus it is the Examiner's position that Summer's provides support for various classes of data in different secure or isolated settings and that the cycles of executing inherent to a bus executing with respect to these different "classes" of data does not agree with the Appellant's position.

Appellant additionally argues

"there is no hook to connect Summer's teachings with Coulouris and Silberschatz, so the references cannot be properly combined. Furthermore, Summers does not teach or suggest what the Examiner asserts it does – Summer's "isolation" works differently and incompatibly to the claimed invention.

The Examiner contends against the Appellant's assertion that "there is no Hook to combine Summer's teachings with Coulouris and Silberschatz." As the Appellant recites on page 4, Coulouris and Silberschatz is a college level textbook disclosing various software techniques. It is understood by those of ordinary skill in the art that the specific hardware implementation is not disclosed because its realization is within the knowledge of an ordinary engineer in his/her art. Summer's broadly provides an example implementation of a secure bus which is able to maintain a distinction between different classes of data, wherein such classes of data may be deemed secure. Thus the different classes of data relied upon in Coulouris and Silberschatz may be compatibly realized in the secure arbiter hardware of Summer's.

In reference to claim 9:

Takahashi does not teach any sort of page table map or a selection unit to select which page table map is applied response to receipt of an event. Instead Takahashi's processor executes instructions from an internal read-only memory ("ROM") when it is in secure mode, and executes instructions from an external random external memory ("RAM") when in general mode. A hardware control circuit prevents the execution of RAM instructions in secure mode, but this is different from selecting one of two page maps.

Furthermore, Takahashi lacks any sort of isolated execution circuit to generate isolated access bus cycles if the apparatus operates in an isolated execution mode. (In Takahashi secure mode, RAM can still be accessed to obtain data, but no distinct type of bus cycle is described or implied.) again, the Examiner turns to Summers for its isolated of untrusted

peripherals, but as explained above, the secondary reference does not generate or use isolated bus cycles either.

The Examiner contends however that Takahashi maintains a dual mode processor with which it accesses various secure primitives that may be broadly construed as page table.

Figure 5, discloses a block of memory where various parts of the memory are reserved for different functions. For example, the first part of the memory holds the secure mode entry routine, while the addresses following it holds the internal pointers to secure functions, and the secure functions themselves. The Examiner contends that the disclosure of Takahashi to execute these instructions from a memory may be broadly construed as a page table map.

As argued above, Summer's discloses isolated bus cycles in that it recites a secure bus arbiter capable of distinguishing between different classes of data, one of which may be secure data, while the bus cycle is merely the unit of activity to indicate a single quantum of execution characteristic of all standard buses.

bus cycle (′ bəs ′ sī·kəl)

(computer science) A single transaction between the main memory and the CPU.

www.answers.com

Furthermore, the Examiner contends that it is clear Takahashi discloses various isolated execution modes. (Column 3, lines 25-40, 45-60) & (Column 4, lines 23-60)

Thus explicit reliance on Summer's to recite the isolated context in which a bus is to function is not necessary as Takahashi discloses the environment for it in the citations given.

In reference to claim 12:

"...Takahashi lacks control registers to define a current memory map, or anything remotely like such registers; indeed, it lacks even the concept of a memory map. Furthermore, even assuming (solely for the sake of argument) that Takahashi's RAM and ROM could reasonably be described as different memory maps, neither of the references teach or suggest an isolated execution circuit to generate isolated access bus cycles."

The Examiner contends that the control registers to define a memory map are inherent to the functioning of a memory map. A control register to define a current memory map is merely an area where the addresses of a RAM may be temporarily stored. Simply put, if a computer may access a memory by address, then it must also have the means to store that address if temporarily. (which is the nature of a register)

Finally the Examiner notes that with independent claims 9 and 12

There is no interconnection between the isolated execution circuit and the selection unit to generate the page table maps. That is there is no explicit recitation in the claim that the

selection signal functions in concert with the isolated execution circuit to provide isolated execution. Broadly construed, Appellant has merely claimed an aggregation of individual parts. Thus while the Appellant has made several arguments against the technical feasibility of the combination of the Examiner's references (page 6, paragraphs 1,3 & page 7, paragraph 2), the Examiner contends that claims 9 and 12 do not explicitly recite an innerworking necessary for the Examiner to meet in the 103 rejections of the claims.

The Appellant has also not recited an explicit recitation of what he or she means by the term "isolated." While it appears from the various arguments that Appellant desires a different interpretation of "isolation" from how the Examiner is interpreting it(arguments page 6, paragraph 3), lack of an explicit recitation of the comprising elements to meet the definition of "isolation" has left the Examiner to broadly construe this term.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Thomas M Ho AU 2132
Thomas M Ho
AU 2132

Gilberto Barron
GILBERTO BARRON JR
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Conferees:

Gilberto Barron GB

Benjamin Lanier BL